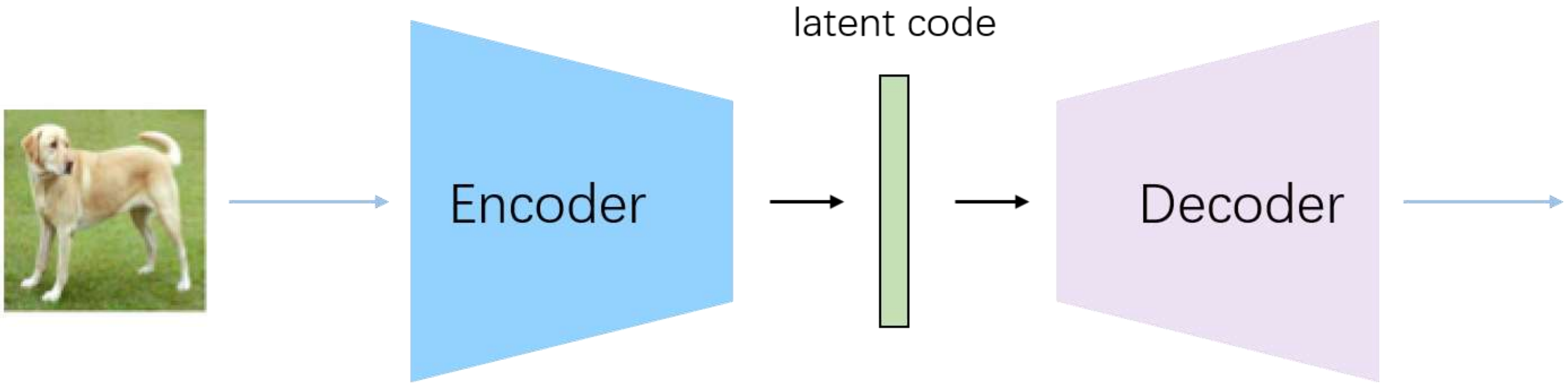
Generation & Understanding

牛宇威

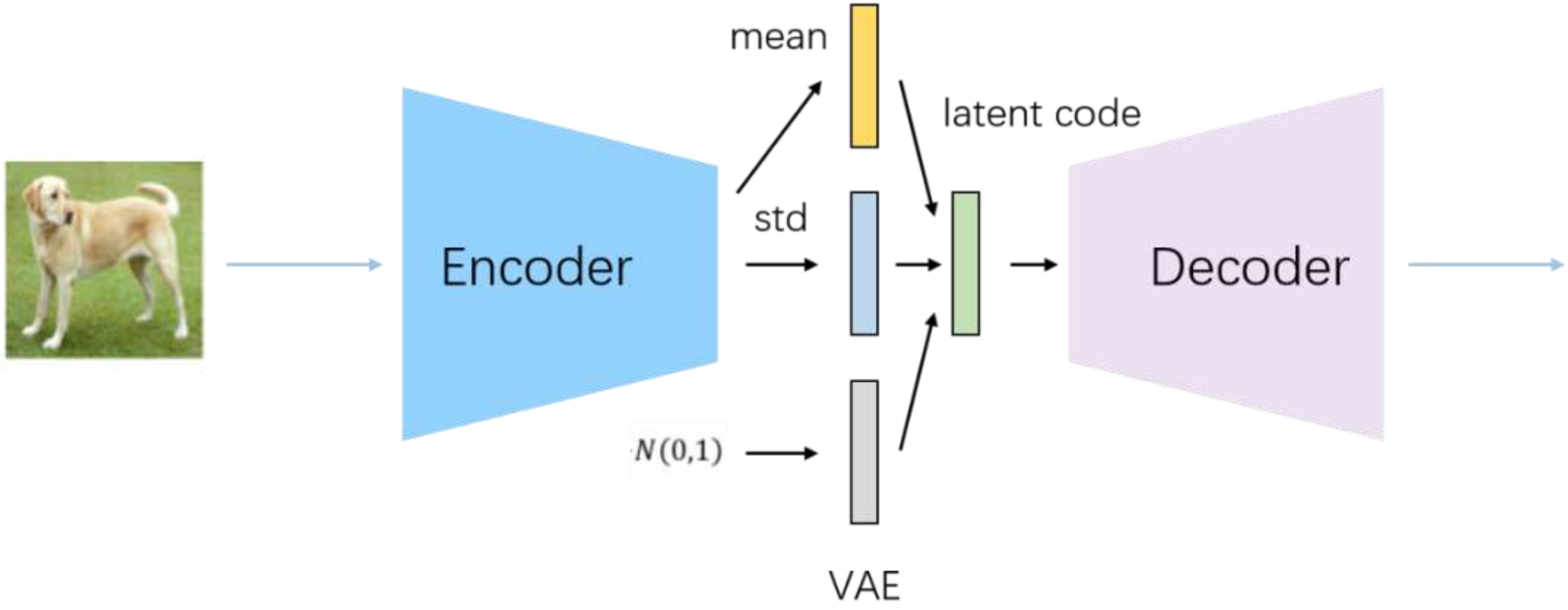
2024/12/6

AE

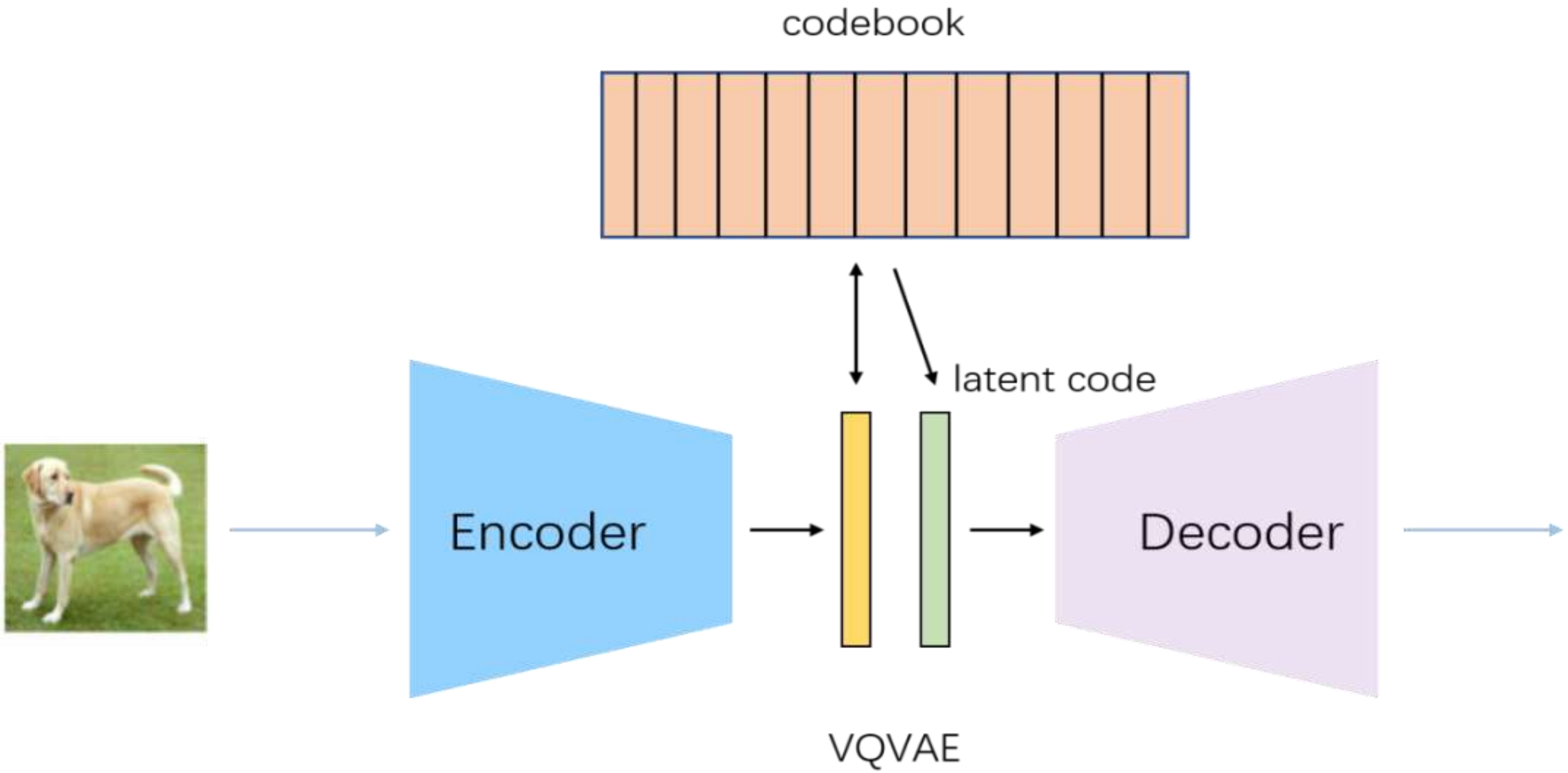


AutoEncoder

VAE



VQ-VAE



Autoregressive Image Generation without Vector Quantization

Tianhong Li[1] Yonglong Tian[2] He Li[3] Mingyang Deng[1] Kaiming He[1]

[1]MIT CSAIL [2]Google DeepMind [3]Tsinghua University

自回归生成图片的两个问题：
1.文本是一维的，天然有先后顺序以供自回归生成。而图像是二维的，没有先后顺序。
2.图像的颜色值是连续而非离散的。而只有离散值才能用类别分布表示。

如果要去除VQ，要怎么样：
1. 找到比VQ更nb的连续变离散方法
2. 不用类别分布来建模下一项数据

[2406.11838] Autoregressive Image Generation without Vector Quantization

https://zhouyifan.net/2024/07/27/20240717-ar-wo-vq/

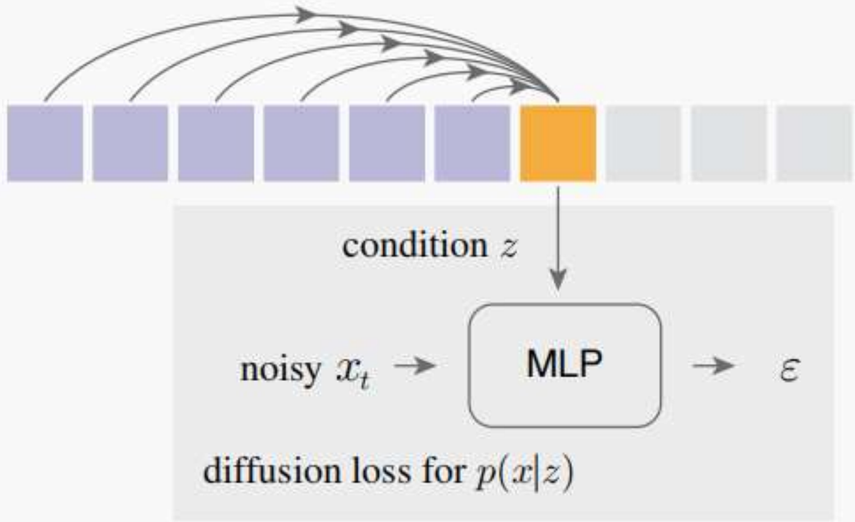


Figure 1: **Diffusion Loss**. Given a continuous-valued token x to be predicted, the autoregressive model produces a vector z, which serves as the condition of a denoising diffusion network (a small MLP). This offers a way to model the probability distribution $p(x|z)$ of *this token*. This network is trained jointly with the autoregressive model by backpropagation. At inference time, with a predicted z, running the reverse diffusion procedure can sample a token following the distribution: $x \sim p(x|z)$. This method eliminates the need for discrete-valued tokenizers.

Condition：上下文像素过transformer的输出
训练一个带有condition的极简DDPM
VAE(KL16) Encoder—Transformer—diffusion建模自回归-
-VAE(KL16) Decoder

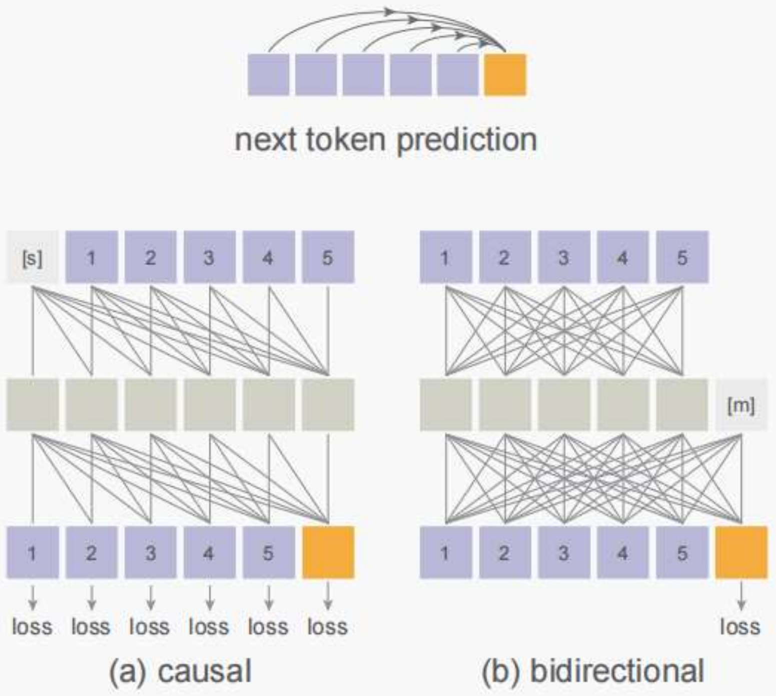


Figure 2: **Bidirectional attention can do autoregression**. In contrast to conventional wisdom, the broad concept of "autoregression" (next token prediction) can be done by either causal or bidirectional attention. (a) **Causal** attention restricts each token to attend only to current/previous tokens. With input shifted by one start token `[s]`, it is valid to compute loss on *all* tokens at training time. (b) **Bidirectional** attention allows each token to see *all* tokens in the sequence. Following MAE [21], mask tokens `[m]` are applied in a middle layer, with positional embedding added. This setup only computes loss on unknown tokens, but it allows for full attention capabilities across the sequence, enabling *better* communication across tokens. This setup can generate tokens one by one at inference time, which is a form of autoregression. It also allows us to predict multiple tokens simultaneously.

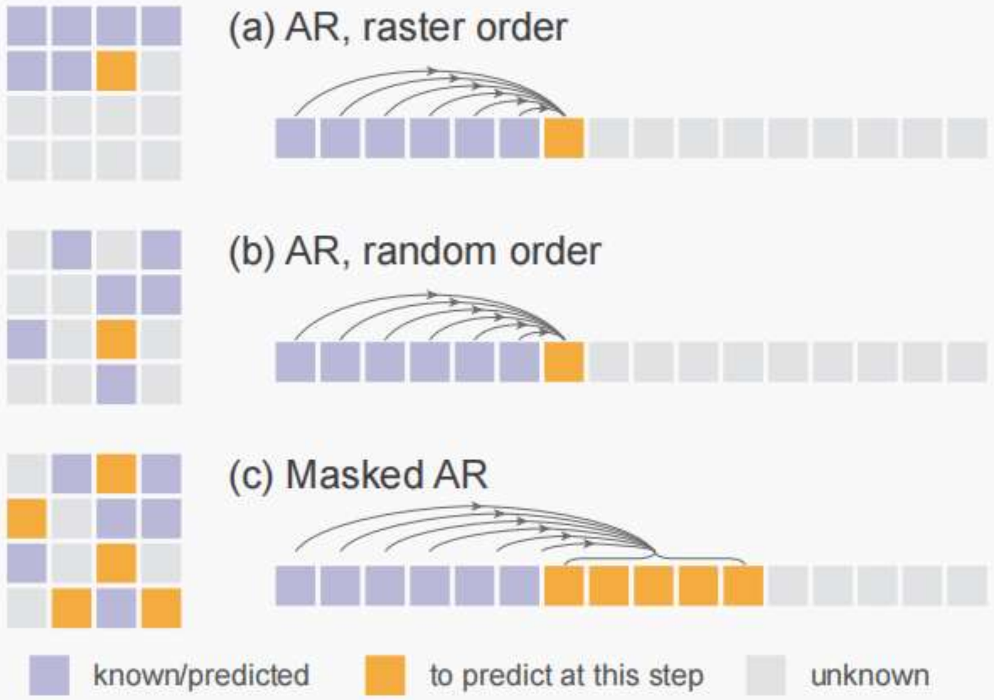
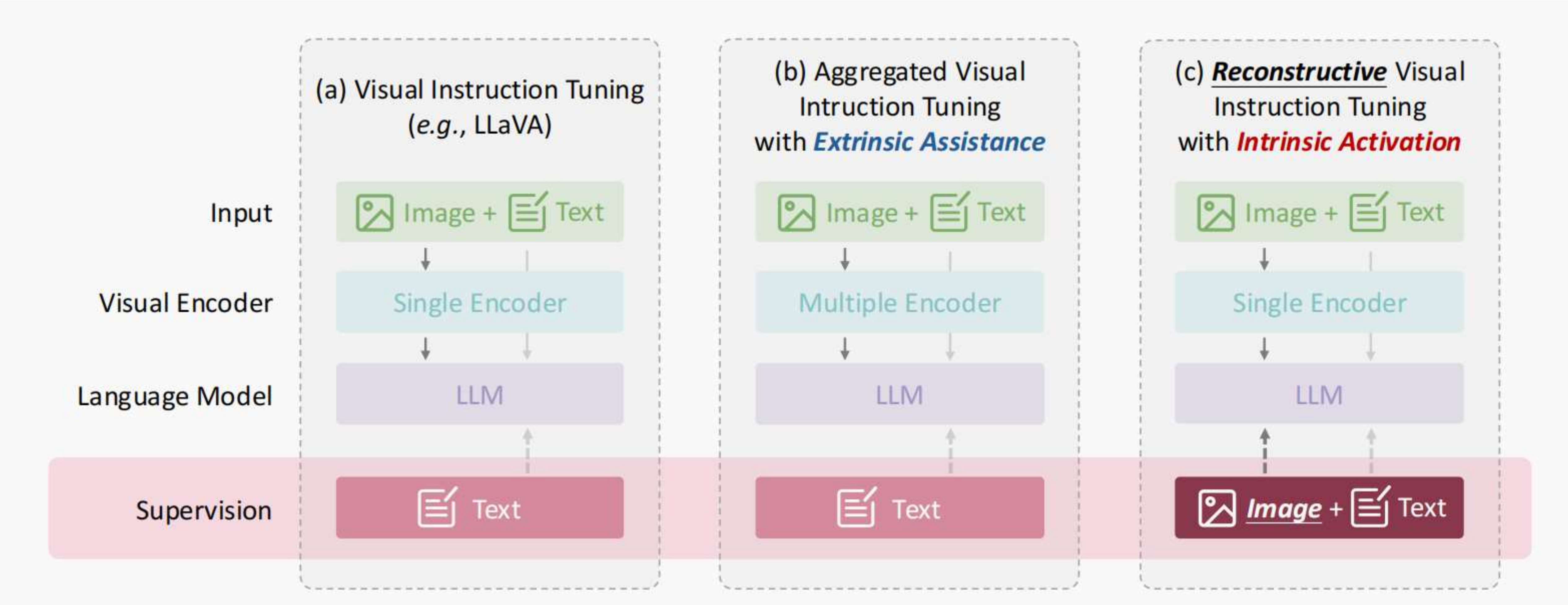
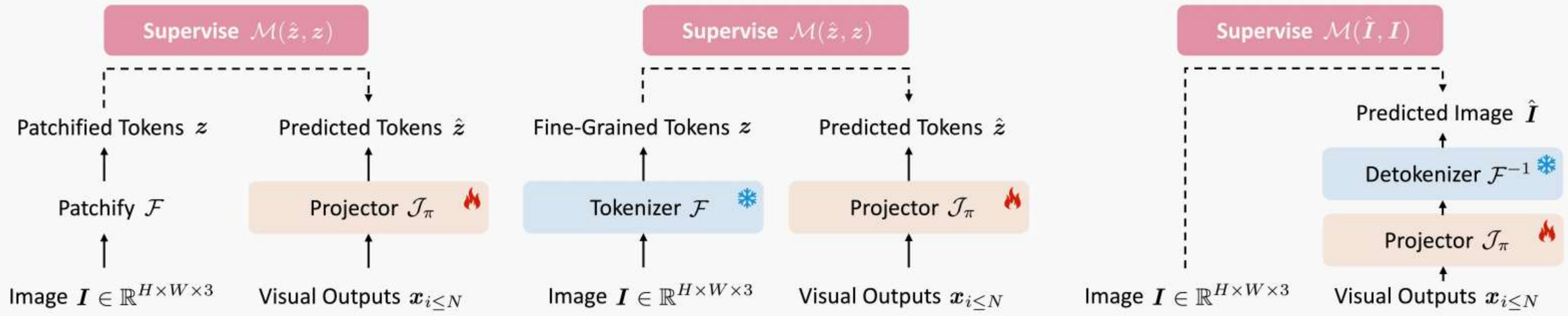


Figure 3: **Generalized Autoregressive Models.** (a) A standard, raster-order autoregressive model predicts one next token based on the previous tokens. (b) A random-order autoregressive model predicts the next token given a random order. It behaves like randomly masking out tokens and then predicting one. (c) A Masked Autoregressive (MAR) model predicts multiple tokens simultaneously given a random order, which is conceptually analogous to masked generative models [4, 29]. In all cases, the prediction of one step can be done by causal or bidirectional attention (Figure 2).

Reconstructive Visual Instruction Tuning

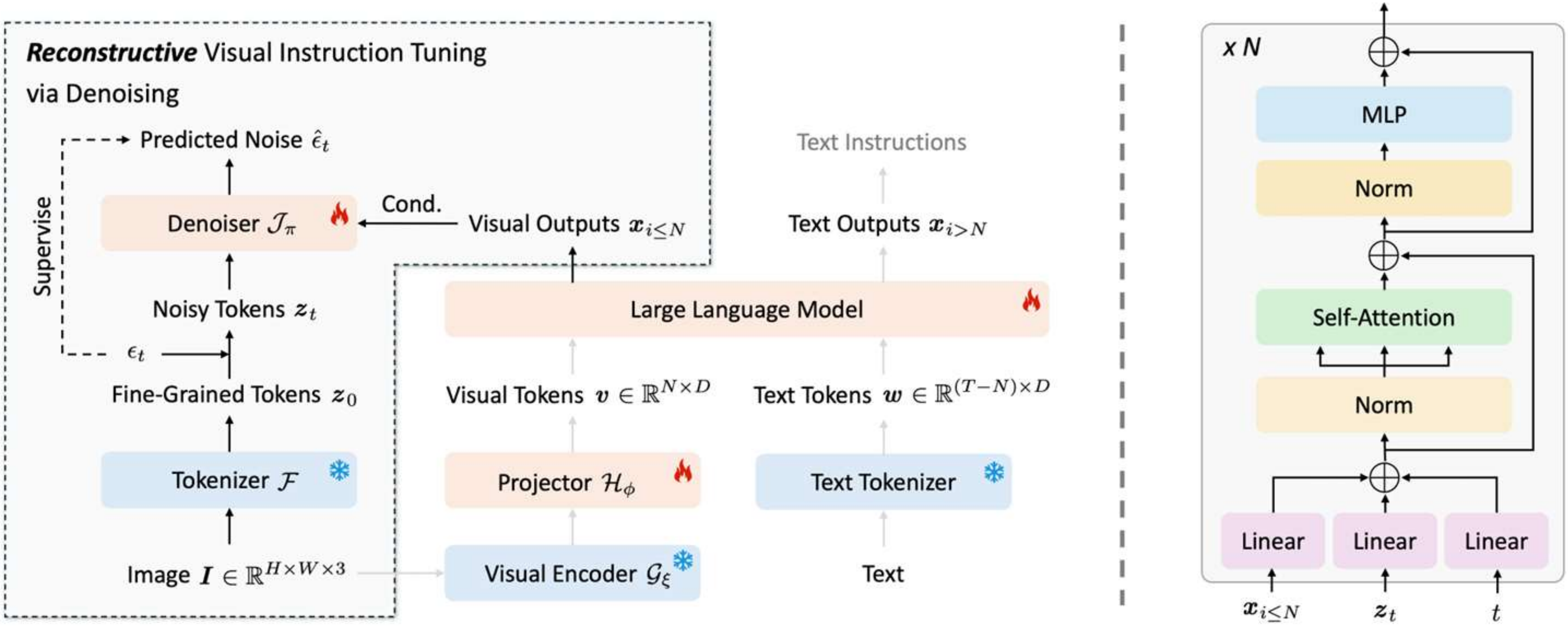


[2410.09575] Reconstructive Visual Instruction Tuning



(a) **ROSS**$^{\mathrm{R}}$-Pixel (b) **ROSS**$^{\mathrm{R}}$-Latent (c) **ROSS**$^{\mathrm{R}}$-Latent2Pixel

Figure 3: Variants of **ROSS**$^{\mathrm{R}}$, where *regression* objectives are either computed on raw RGB values in (a) and (c), or specific latent space determined by $\mathcal{F}$ in (b). We adopt MSE as $\mathcal{M}$ for *pixel* regression in (a) and (c), and cosine-similarity for *latent* regression in (b), respectively.



(a) The framework of **ROSS**$^{\mathrm{D}}$. (b) The denoiser.

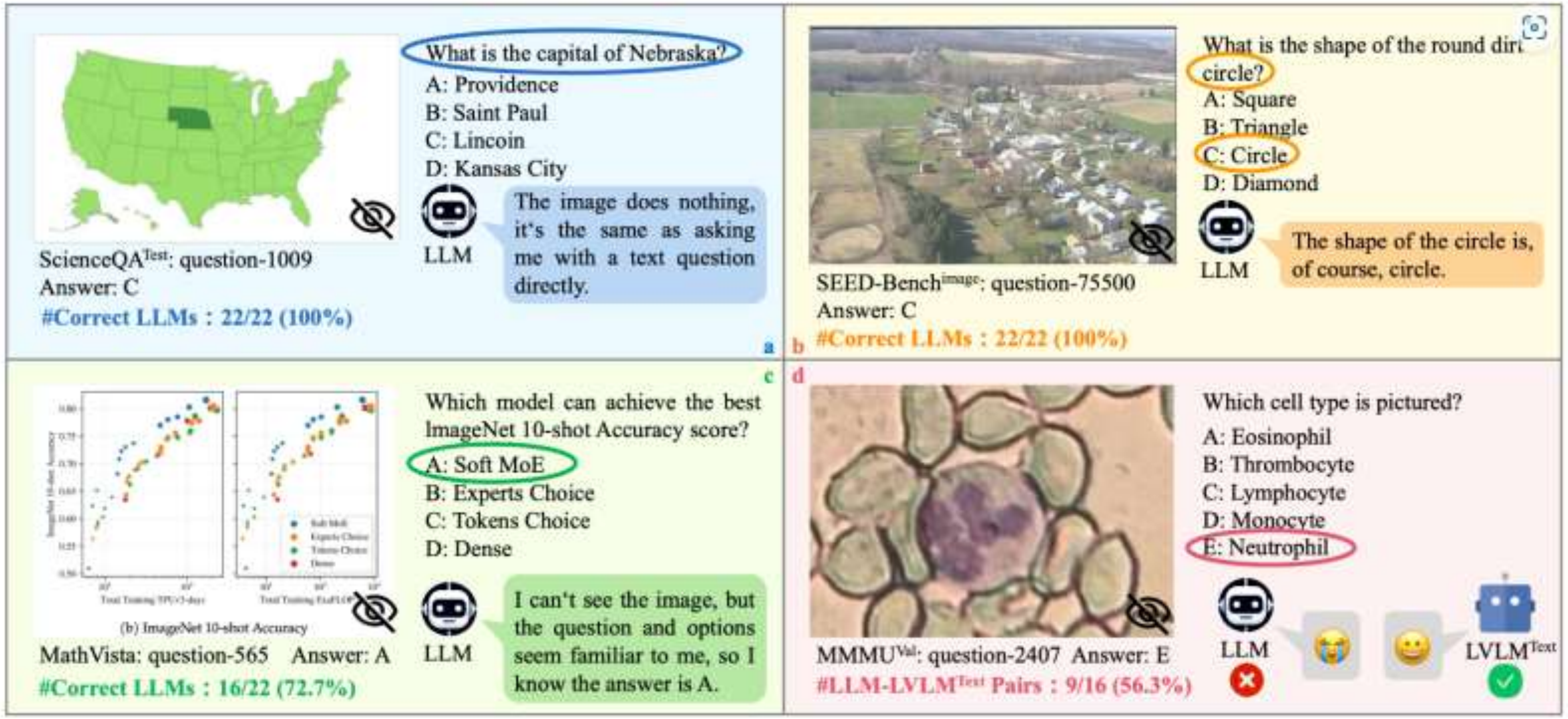
Figure 4: Illustration of (a) the training procedure of **ROSS**$^{\mathrm{D}}$ and (b) the detailed architecture of the denoiser $\mathcal{J}_\pi$. (a) **ROSS**$^{\mathrm{D}}$ introduces visual guidance via *denoising fine-grained visual tokens* $\boldsymbol{z}_0$ *conditioning on visual outputs* $\boldsymbol{x}_{i\leq N}$. (b) The denoiser takes noisy tokens $\boldsymbol{z}_t$, current timesteps t, and conditions $\boldsymbol{x}_{i\leq N}$ as inputs and outputs the predicted noise $\hat{\epsilon}_t$. Each denoiser block consists of three linear projection layers and a standard self-attention block (Vaswani et al., 2017).

Benchmark	CLIP				SigLIP			
LLM	Vicuna		Qwen2		Vicuna		Qwen2	
Method	LLaVA	Ross	LLaVA	Ross	LLaVA	Ross	LLaVA	Ross
POPE-acc	86.3	**87.2** ↑ **0.9**	87.9	**88.4** ↑ **0.5**	86.0	**87.7** ↑ **1.7**	88.5	**88.7** ↑ **0.2**
HallusionBench-aAcc	52.5	**55.8** ↑ **3.3**	55.0	**59.1** ↑ **4.1**	50.4	**53.8** ↑ **3.4**	57.3	**58.2** ↑ **0.9**
MMBench-EN-dev	67.0	**67.6** ↑ **0.6**	73.8	**75.2** ↑ **1.4**	64.5	**69.2** ↑ **4.7**	76.3	**76.9** ↑ **0.6**
MMBench-CN-dev	**60.0**	59.8 ↓ 0.2	72.9	**73.7** ↑ **0.8**	63.1	**63.4** ↑ **0.3**	75.7	**76.3** ↑ **0.7**
SEED-img	**66.7**	66.4 ↓ 0.3	70.3	**70.7** ↑ **0.4**	68.2	**69.0** ↑ **0.8**	**72.3**	72.1 ↓ 0.2
MMMU-dev	30.0	**34.0** ↑ **4.0**	44.0	**45.3** ↑ **1.3**	33.3	**38.0** ↑ **4.7**	38.7	**41.3** ↑ **2.6**
MMMU-val	35.3	**36.0** ↑ **0.7**	41.9	**42.6** ↑ **0.7**	34.2	**35.4** ↑ **1.2**	41.8	**43.8** ↑ **2.0**
MMVP	28.0	**36.3** ↑ **8.3**	29.6	**42.2** ↑ **12.6**	27.3	**38.0** ↑ **10.7**	40.7	**49.3** ↑ **8.6**
AI2D-test	61.2	**61.4** ↑ **0.2**	71.9	**73.3** ↑ **1.4**	**62.6**	62.4 ↓ 0.2	74.0	**74.5** ↑ **0.5**
ChartQA-test	32.9	**39.8** ↑ **6.9**	36.2	**41.6** ↑ **5.4**	34.0	**48.2** ↑ **14.2**	44.4	**46.9** ↑ **2.5**
DocVQA-val	33.4	**41.6** ↑ **8.2**	31.1	**44.7** ↑ **13.6**	40.4	**40.7** ↑ **0.3**	39.2	**39.3** ↑ **0.1**
InfoVQA-val	21.2	**26.4** ↑ **5.2**	22.1	**39.3** ↑ **16.2**	22.8	**23.3** ↑ **0.5**	24.0	**25.1** ↑ **1.1**
TextVQA-val	55.7	**58.7** ↑ **3.0**	52.0	**54.1** ↑ **2.1**	60.5	**62.6** ↑ **2.1**	56.3	**57.5** ↑ **1.2**
OCRBench	339	**350** ↑ **11**	363	**381** ↑ **18**	354	**365** ↑ **11**	432	**448** ↑ **16**
RealWorldQA	52.7	**53.2** ↑ **0.5**	56.7	**57.4** ↑ **0.7**	55.0	**57.1** ↑ **2.1**	57.9	**59.1** ↑ **1.2**
Average	47.8	**50.6** ↑ **2.8**	52.1	**56.4** ↑ **4.3**	49.2	**52.4** ↑ **3.2**	55.4	**56.9** ↑ **1.5**

CLIP: CLIP-ViT-L/14@336; SigLIP: SigLIP-SO400M-ViT-L/14@384; Vicuna: Vicuna-7B-v1.5; Qwen2: Qwen2-7B-Instruct

THE OVERLOOKED ISSUES FOR EVALUATING LVLMS

We highlight cases in existing multi-modal benchmarks where evaluation samples either lack visual dependency or have unintentionally leaked into the training data of LLMs and LVLMs.



(a) Some samples can be answered by LLMs using only text-based world knowledge;

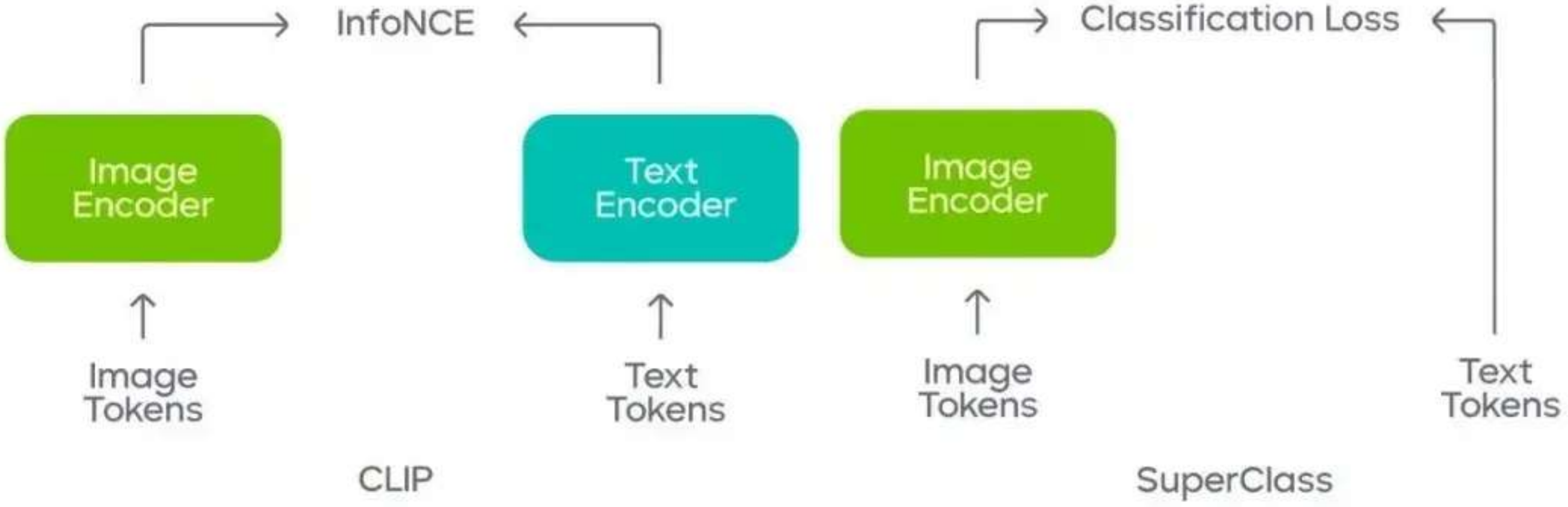
(b) For some instances, the question itself contains the answer, making images superfluous;

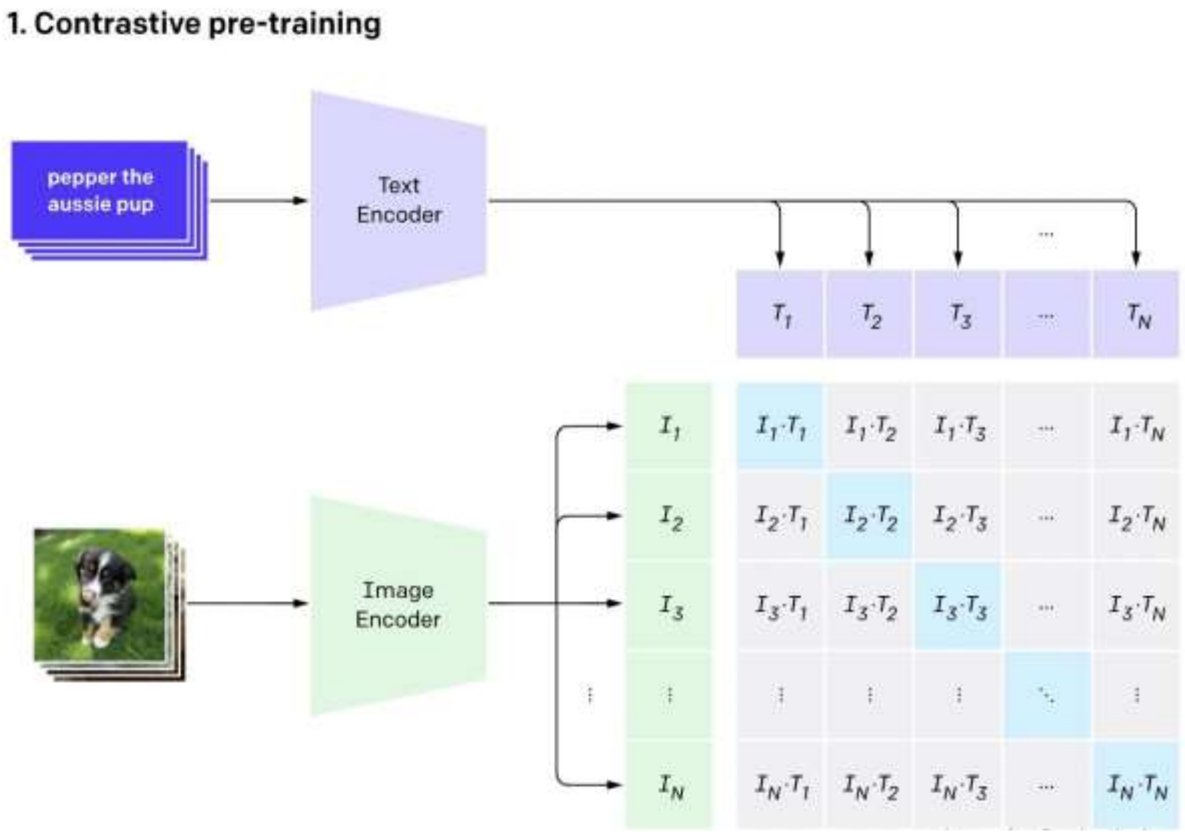
(c) Some samples are leaked into LLMs' training corpora can be "recalled" with the textual questions and answers directly;

(d) Some samples indiscernible to LLMs but solved by LVLMs without accessing images suggest leakage into LVLMs' multi-modal training data.

Classification Done Right for Vision-Language Pre-Training

Zilong Huang Qinghao Ye Bingyi Kang Jiashi Feng Haoqi Fan
ByteDance Research

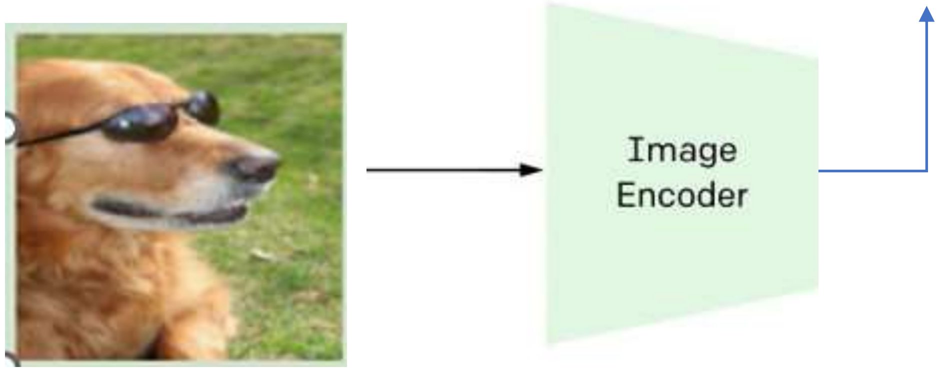




A golden retriever wearing sunglasses

[0,…,0,1,0,…0,1,0,…,0,1,0,…, 0,…,0,1,0 0,…,0,1,0,…,0]

Classification Loss



[2411.03313] Classification Done Right for Vision-Language Pre-Training

Table 1: Comparison of the Linear probing top-1 accuracy on ImageNet-1K dataset.

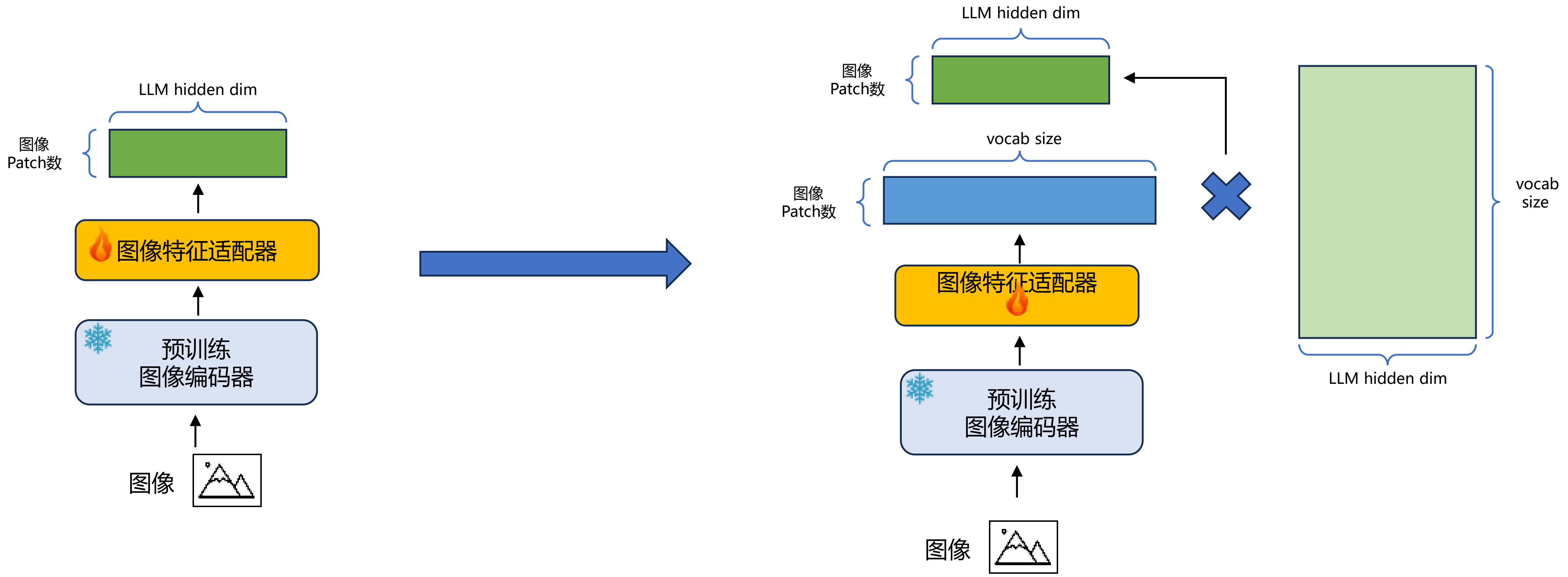
Method	PreTraining data	ViT-Base		ViT-Large	
		#Seen Samples	Top-1 (%)	#Seen Samples	Top-1 (%)
contrastive or clustering based					
MoCov3 [10]	IN1K	400M	76.7	400M	77.6
DINO [5]	IN1K	512M	78.2	-	-
iBOT [80]	IN22K	400M	79.5	256M	81.0
DINOv2 [55]	LVD-142M	-	-	2B	84.5
reconstruction based					
BEiT [3]	D250M+IN22K	1B	56.7	1B	73.5
SimMIM [73]	IN1K	1B	56.7	-	-
CAE [8]	D250M	2B	70.4	2B	78.1
MAE [24]	IN1K	2B	68.0	2B	75.8
vision-language pretraining based					
Openai CLIP [57]	WIT-400M	13B	78.5	13B	82.7
Cappa [70]	WebLI-1B	-	-	9B	83.0
OpenCLIP [29]	Datacomp-1B	-	-	13B	83.9
SuperClass	Datacomp-1B	1B	78.7	1B	82.6
SuperClass	Datacomp-1B	13B	**80.2**	13B	**85.0**

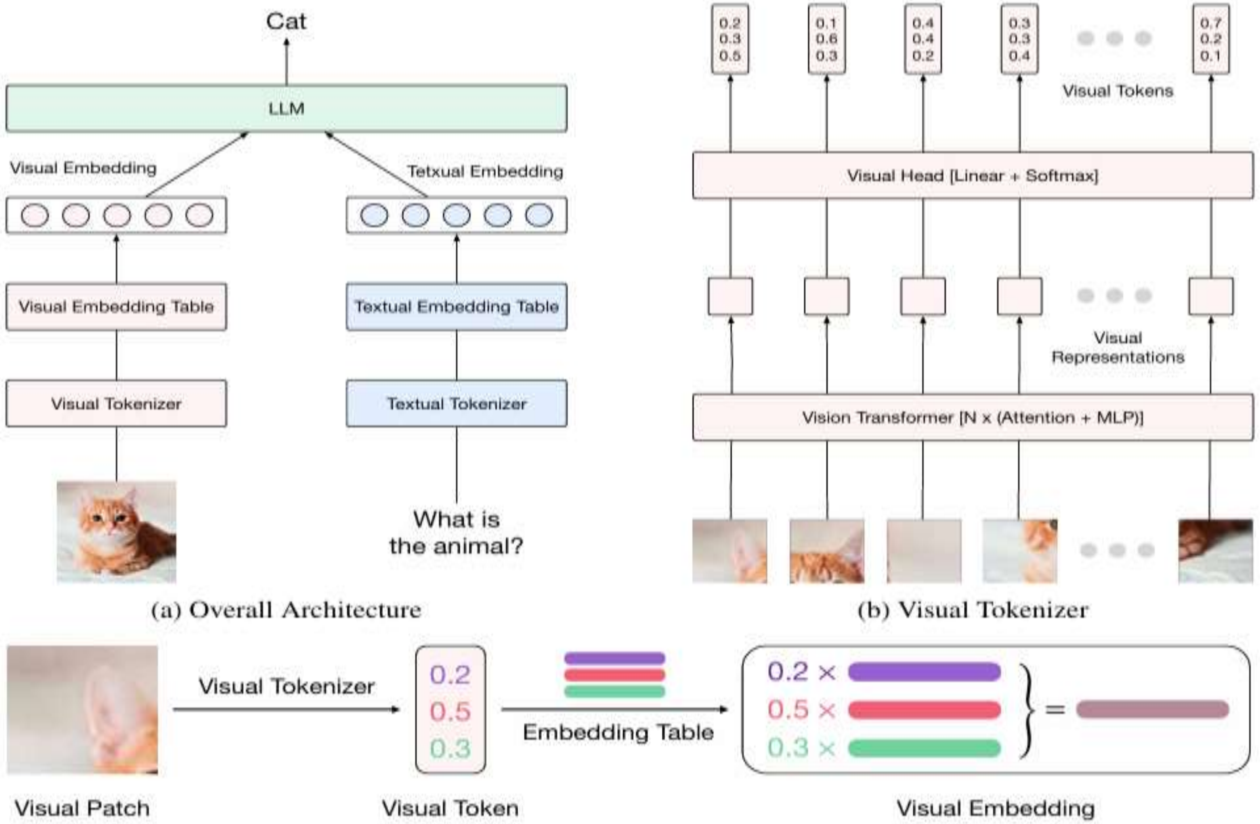
Table 11: The performance of vision & language downstream tasks with different pretrained models.

Method	VQAv2	GQA	VizWiz	T-VQA	SciQA	MME	MMB	PoPE	MMMU
OpenCLIP	74.54	61.03	50.47	38.16	**67.33**	**1434**/269	60.73	85.52	35.9
MAE	63.50	54.58	50.22	11.55	54.75	1175/343	42.44	80.69	35.7
DINOv2	73.32	**61.87**	49.15	14.08	64.90	1336/297	57.90	**86.24**	35.3
SuperClass	**75.24**	60.96	**54.33**	**39.20**	66.09	1371/**322**	**63.14**	85.69	**36.0**

Method

将适配器输出特征约束为Language Basis Vector的线性系数
将visual embedding约束在Language Basis Vector的线性子空间





(a) Overall Architecture

(b) Visual Tokenizer

(c) From visual patch to visual embedding

Figure 3: Illustration of Ovis. Figure (a) shows the whole architecture of Ovis, which contains two embedding tables for visual and textual inputs. Figure (b) illustrates how a visual patch is first mapped to a probabilistic token. Figure (c) demonstrates that the probabilistic token helps select multiple embeddings from the embedding table and output their weighted combination.

Ovis: Structural Embedding Alignment for Multimodal Large Language Mode